package com.twitter.follow\_recommendations.configapi.common

import com.twitter.timelines.configapi.FeatureSwitchOverrideUtil.DefinedFeatureName

import com.twitter.timelines.configapi.FeatureSwitchOverrideUtil.ValueFeatureName

import com.twitter.timelines.configapi.BoundedParam

import com.twitter.timelines.configapi.FSBoundedParam

import com.twitter.timelines.configapi.FSName

import com.twitter.timelines.configapi.HasDurationConversion

import com.twitter.timelines.configapi.OptionalOverride

import com.twitter.timelines.configapi.Param

import com.twitter.util.Duration

trait FeatureSwitchConfig {

def booleanFSParams: Seq[Param[Boolean] with FSName] = Nil

def intFSParams: Seq[FSBoundedParam[Int]] = Nil

def longFSParams: Seq[FSBoundedParam[Long]] = Nil

def doubleFSParams: Seq[FSBoundedParam[Double]] = Nil

def durationFSParams: Seq[FSBoundedParam[Duration] with HasDurationConversion] = Nil

def optionalDoubleFSParams: Seq[

(BoundedParam[Option[Double]], DefinedFeatureName, ValueFeatureName)

] = Nil

def stringSeqFSParams: Seq[Param[Seq[String]] with FSName] = Nil

/\*\*

\* Apply overrides in list when the given FS Key is enabled.

\* This override type does NOT work with experiments. Params here will be evaluated for every

\* request IMMEDIATELY, not upon param.apply. If you would like to use an experiment pls use

\* the primitive type or ENUM overrides.

\*/

def gatedOverridesMap: Map[String, Seq[OptionalOverride[\_]]] = Map.empty

}

object FeatureSwitchConfig {

def merge(configs: Seq[FeatureSwitchConfig]): FeatureSwitchConfig = new FeatureSwitchConfig {

override def booleanFSParams: Seq[Param[Boolean] with FSName] =

configs.flatMap(\_.booleanFSParams)

override def intFSParams: Seq[FSBoundedParam[Int]] =

configs.flatMap(\_.intFSParams)

override def longFSParams: Seq[FSBoundedParam[Long]] =

configs.flatMap(\_.longFSParams)

override def durationFSParams: Seq[FSBoundedParam[Duration] with HasDurationConversion] =

configs.flatMap(\_.durationFSParams)

override def gatedOverridesMap: Map[String, Seq[OptionalOverride[\_]]] =

configs.flatMap(\_.gatedOverridesMap).toMap

override def doubleFSParams: Seq[FSBoundedParam[Double]] =

configs.flatMap(\_.doubleFSParams)

override def optionalDoubleFSParams: Seq[

(BoundedParam[Option[Double]], DefinedFeatureName, ValueFeatureName)

] =

configs.flatMap(\_.optionalDoubleFSParams)

override def stringSeqFSParams: Seq[Param[Seq[String]] with FSName] =

configs.flatMap(\_.stringSeqFSParams)

}

}